IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

HEADWATER RESEARCH LLC, *Plaintiff*, v. VERIZON COMMUNICATIONS INC., CELLCO PARTNERSHIP d/b/a VERIZON WIRELESS, and VERIZON CORPORATE SERVICES GROUP, INC., *Defendants*.	§	Case No. 2:23-cv-00352-JRG-RSP

REPORT AND RECOMMENDATION

Before the Court is the Motion for Summary Judgment of Non-Infringement of U.S. Patent No. 8,589,541, filed by Defendants Verizon Communications Inc., Cellco Partnership d/b/a Verizon Wireless, and Verizon Corporate Services Group, Inc.. **Dkt. No. 177**. For the reasons discussed below, the Motion should be **DENIED**.

I. BACKGROUND

On July 28, 2023, Plaintiff Headwater Research LLC filed suit against Defendants, asserting that they infringe, *inter alia*, U.S. Patent No. 8,589,541. Dkt. No. 1 at 1. The '541 Patent, entitled "Device-Assisted Services for Protecting Network Capacity," contains one independent claim and 173 dependent claims. *See* '541 Patent. Headwater alleges that Defendants infringe dependent Claims 79 and 83, which both depend from Claim 1. Dkt. Nos. 177 at 2; 177-2 at ¶ 22; 209 at 2. Claim 1 recites machine-executable instructions that, when executed by one or more processors of a wireless end-user device, cause the processor(s) to:

> ***identify a service usage activity of the wireless end-user device***, the service usage activity being associated with a first software component of a plurality of software components on the wireless end-user device, the service usage activity comprising one or more prospective or successful communications over a wireless network; ***determine whether the service usage activity comprises a background activity***;

> determine at least an aspect of a policy based on a user input obtained through a user interface of the wireless end-user device or based on information from a network element, the policy to be applied if the service usage activity is the background activity, the policy at least for controlling the service usage activity; and ***if it is determined that the service usage activity is the background activity, apply the policy***.

'541 Patent at Claim 1 (emphasis added).

In the Court's Claim Construction Order, the Court gave "service usage activity" its plain and ordinary meaning, though it noted that "service usage activity" cannot be "an application or software component." Dkt. No. 140 at 12, 21. Asserted Claim 79 adds the limitation "wherein apply the policy comprises at least assist in ***intercepting*** a stack application programming interface (API) level or application messaging layer request." '541 Patent at Claim 79 (emphasis added). Asserted Claim 83 adds the limitation "wherein apply the policy comprises at least assist in ***intercepting***, modifying, blocking, removing, injecting, swapping, or replacing an application interface message." '541 Patent at Claim 83 (emphasis added).

On April 2, 2025, Defendants filed the instant Motion, alleging that summary judgment of non-infringement for the '541 Patent is warranted for two independent reasons. Dkt. 177 at 1-2. First, Defendants contend that summary judgment is warranted because the accused products do not satisfy Claim 1 as the accused products never examine a service usage activity to determine that it is background activity and never apply a policy based on that determination. *Id.* Second, Defendants contend that summary judgment is warranted because the accused products do not meet the "intercepting" or disrupting limitations of Claims 79 and 83. *Id.*

II. APPLICABLE LAW

Summary judgment should be granted "if the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law." Fed. R. Civ. P. 56(a). Any evidence must be viewed in the light most favorable to the nonmovant. *See Anderson*

v. Liberty Lobby, Inc., 477 U.S. 242, 255 (1986) (citing *Adickes v. S.H. Kress & Co.*, 398 U.S. 144, 158–59 (1970)). Summary judgment is proper when there is no genuine dispute of material fact. *Celotex v. Catrett*, 477 U.S. 317, 322 (1986). "By its very terms, this standard provides that the mere existence of some alleged factual dispute between the parties will not defeat an otherwise properly supported motion for summary judgment; the requirement is that there be no genuine [dispute] of material fact." *Anderson*, 477 U.S. at 247–48. The substantive law identifies the material facts, and disputes over facts that are irrelevant or unnecessary will not defeat a motion for summary judgment. *Id.* at 248. A dispute about a material fact is "genuine" when the evidence is "such that a reasonable jury could return a verdict for the nonmoving party." *Id.*

The moving party must identify the basis for granting summary judgment and evidence demonstrating the absence of a genuine dispute of material fact. *Celotex*, 477 U.S. at 323. If the moving party does not have the ultimate burden of persuasion at trial, the party 'must either produce evidence negating an essential element of the nonmoving party's claim or defense or show that the nonmoving party does not have enough evidence of an essential element to carry its ultimate burden of persuasion at trial.'" *Intellectual Ventures I LLC v. T Mobile USA, Inc*., No. 2:17-CV-00577-JRG, 2018 WL 5809267, at *1 (E.D. Tex. Nov. 6, 2018) (quoting *Nissan Fire & Marine Ins. Co., Ltd. v. Fritz Cos., Inc.*, 210 F.3d 1099, 1102 (9th Cir. 2000)).

III. ANALYSIS

Defendants contend that summary judgment of non-infringement is warranted because the accused products neither meet the limitations of Claim 1 nor the "intercepting" or otherwise disrupting limitations in Claims 79 and 83. Dkt. No. 177 at 1-2. The Court will begin with Defendants' argument as to the limitations in Claim 1.

A. Independent Claim 1

Defendants argue that the accused products do not satisfy the limitations in Claim 1 because they do not manage background activity by looking at a particular service usage activity, they do not determine if a communication is a background activity, nor do they apply a policy based on whether a particular communication is a background activity. Dkt. No. 177 at 7. Instead, Defendants argue that the accused Apple and Android products function by "request[ing] pre-clearance to engage in background communications." *Id.*

1. The Accused Apple Products

In the Apple Accused Products, which run iOS, Defendants contend that background activities are not managed by looking at any "service usage activity" to determine if it is a background activity. Dkt. No. 177 at 7. Instead, Defendants contend that before an app can engage in background activities, the app must first send a request to the iOS scheduler through a submitTaskRequest API call to be allotted a window of time, during which the app will be awakened from its suspended state and allowed to run, if Background App Refresh is enabled. *Id.* Defendants contend that identifying a "task request" to iOS cannot satisfy Claim 1, however, because it is not "service usage activity" due to it being "purely internal to the device software" and not including "prospective or successful communications over a wireless network." *Id.* at 8.

Headwater responds that Dr. Wesel mapped the "service usage activity" in Claim 1 to Background App Refresh *tasks*, which are distinct from Background App Refresh *task requests*. Dkt. No. 209 at 7 (citing *e.g.*, Wesel Rpt. ¶ 435 ("The Apple Accused Products identify a service usage activity of the wireless end-user device. For example, as shown below, the Apple Accused Products identify Background App Refresh *task requests* made by applications running on the devices. Background App Refresh *tasks* are 'service usage activity of the wireless end-user

device.'") (emphasis added)). Headwater states that it is those Background App Refresh *tasks*, rather than the *task requests*, which must be evaluated to determine whether they are "prospective or successful communications over a wireless network." *Id.* (citing *e.g.*, Wesel Rpt. ¶ 826 ("Background App Refresh *tasks* are 'service usage activity of the wireless end-user device' . . . . They also comprise 'one or more prospective or successful communications over a wireless network,' as they are tasks which the operating system understands are ones which require network access . . . .") (emphasis added)).

Defendants contend that two problems exist with Headwater's theory. Dkt. No. 238 at 3. First, Defendants contend that Dr. Wesel only identified the Background App Refresh *task request*, not the Background App Refresh *task* itself, as satisfying the "identify" step in Claim 1. *Id.* at 2-3. Headwater responds that Dr. Wesel identifies the Background App Refresh *task* several times in his report. Dkt. No. 258 at 1-2 (citing *e.g.*, Wesel Rpt. At ¶ 435 ("[T]he Apple Accused Products identify Background App Refresh *task requests* made by applications running on the devices. Background App Refresh *tasks* are 'service usage activity of the wireless end-user device.'" (emphasis added)); ¶ 630 ("iOS knows which tasks (Background App Refresh tasks or otherwise) are associated with which applications because tasks are associated with unique identifiers, ..."); ¶ 648 ("The developer … adds an identifier to their application's 'permitted background task scheduler identifiers' property list in its info.plist (the 'information property list file') for identifying the background task refresh method to be used in the application.").

Although Dr. Wesel discusses Background App Refresh *tasks* numerous times in the report in conjunction with the limitations in Claim 1, Dr. Wesel appears to begin the relevant portions of his report by expressly mapping the "identify" limitation recited in Claim 1 to the Background App Refresh *task request*. *See* Wesel Rpt. ¶¶ 435, 630, 826, 852. The Court is not convinced,

however, that Dr. Wesel only maps "service usage activity" to the task request or, more precisely, the "submitTaskRequest." In fact, when asked during his deposition whether the "submitTaskRequest function" is the "service usage activity [he's] pointing to," Dr. Wesel responded that "the request isn't the service usage activity that I point to." Dkt. No. 177-5 at 115:20-25. Dr. Wesel makes clear in response to the next question that "the service usage activity that [he's] pointing to is the *task* that is requested by the application." *Id.* at 116:1-8 (emphasis added). The Court understands that the Background App Refresh *tasks* mapped by Dr. Wesel as the "service usage activity" begin with the *task requests* because, as Headwater notes, the task requests cause the creation of the tasks. *See* Dkt. No. 258 at 1; *see also e.g.*, Wesel Rpt. ¶ 435. Given that the Court is required to view all justifiable inferences drawn from the factual record in the light most favorable to the nonmoving party, the Court does not find that Dr. Wesel only identified the Background App Refresh *task request*. *See Adickes v. S. H. Kress & Co.*, 398 U.S. 144, 159 (1970) (ruling that inferences are to be viewed in the light most favorable to the party opposing the motion for summary judgment). Accordingly, the Court does not find that summary judgment of non-infringement is warranted on this point.

Second, Defendants contend that the Background App Refresh *task* is not "service usage activity" just because it "may" require network access. *Id.* at 3. Though Defendants acknowledge that Dr. Wesel opines that "tasks" comprise "one or more prospective or successful communications" because "they are tasks which the operating system understands are ones which require network access," Defendants contend this is insufficient because this does not mean the task itself "is" a communication over a wireless network. *Id.*

Headwater responds that this fails as a matter of logic because if a "task" that a computer plans to do requires network access and the computer understands that the task requires and will

use network access, the task clearly includes a "prospective" network communication and may include a "successful" network communication. Dkt. No. 258 at 2.

The Court finds that there is a dispute of material fact as to whether the Background App Refresh task is "service usage activity" that satisfies the limitations in Claim 1. On the one hand, Defendants contend that the "task" is not "service usage activity" because "it is not a communication over a wireless network. Dkt. No. 238 at 3 (citing *e.g.*, David Chan, Corp. Rep. Dep. Tr., Dkt. No. 238-6 at 209:19-21 ("[D]oes background app refresh know whether a task involves network communications at all? A: No.")). On the other hand, Dr. Wesel opines that the Background App Refresh task is service usage activity. *See e.g.,* Wesel Rpt. ¶ 826 ("[Background App Refresh tasks] are tasks which the operating system understands are ones which require network access through a variety of means, including specifically through DAS BAR Scheduler's reliance on the barActivity.requiresNetwork and barActivity.requiresInexpensiveNetworking variables, which respectively inform the operating system that a BAR task requires a network and a BAR task requires an 'inexpensive' network (i.e., a wifi network). In the case of Background App Refresh tasks, all such tasks are always coded as requiring a network (barActivity.requiresNetwork is always set to YES), making all such tasks 'one or more prospective or successful communications over a wireless network.'"). Consequently, there is a genuine dispute of material fact as to whether the accused Apple products infringe Claim 1.

Accordingly, the Motion for Summary Judgment of Non-Infringement as to the accused Apple products on this basis should be **DENIED**.

2. The Accused Android Products

In the Android Accused Products, Defendants contend that background activities are not managed by looking at any "service usage activity" to determine if it is a background activity. Dkt.

No. 177 at 8. Instead, Defendants contend that before an app can send any service usage activity at all, it must first make an internal request for available networks using the "requestNetwork" API call, and the ConnectivityManager will respond with information about that network if it is permitted to access that network. *Id.* at 8-9. Defendants argue that while Dr. Wesel asserted in his report that this API call and response satisfied Claim 1's requirement to "identify a service usage activity," Dr. Wesel admitted at his deposition that this API call or "requestNetwork function call" is "not the service usage activity that [he is] looking [at] to meet the element." *Id.* at 9; *Compare* Wesel Rpt. at ¶¶ 319, 331, *with* Dkt. No. 177-4 at 70:3-8. Defendants claim that Dr. Wesel had to admit this because the request from the app and the response from the ConnectivityManager are both internal API messages that do not include "prospective or successful communications over a wireless network" as required. *Id.* at 9.

Headwater responds that Dr. Wesel makes clear that Android determines whether a service usage activity is a background activity based on information about the activity, such as the activity itself or information about the application running the service usage activity. Dkt. No. 209 at 8 (citing Wesel Rpt. ¶¶ 903-04, 1004-06). Headwater argues that this "service usage activity" can include, for example, "network requests through ConnectivityManager" or "requests, packets, or data transmitted to/through the firewall." *Id.* (citing Wesel Rpt. ¶¶ 319, 378). Headwater further contends that Dr. Wesel further explains that Android "determines an application and its activities that are not doing anything visible and not running any foreground services is 'background.'" *Id.* (citing Wesel Rpt. ¶ 903). As support, Headwater notes that Dr. Wesel cites testimony from Android engineers that confirm that Android saves battery by blocking network access requests by apps running in the background. *Id.* (citing *e.g.*, Wesel Rpt. ¶ 184 ("The Android Accused Products meet the limitation wherein they determine whether the service usage activity comprises

a background activity. *See, e.g.*, 1/23/2024 Sharkey Tr. at 177:25-178:6 (under certain conditions, 'one functionality of data saver is that it can also block access requests made by apps to access the internet'); . . . 1/23/2024 Sharkey Tr. at 194:9-22 ('one way the system would save battery is to block network access requests by apps running in the background,' . . . ."); *see also* ¶ 905 ("Testimony from Mr. Jeff Sharkey of Android's development team confirms that Android determines which application processes are 'foreground' and which are 'background' at e.g. 1/23/24 Transcript at p. 40. This establishes that the claim limitation is satisfied. Android's determination that an app is a background app by tracking its processes (including its activities) satisfies the requirement to determine if the internet service activity is background activity.")). Headwater also claims that Defendants ignore the full scope of Dr. Wesel's answer at deposition, where he identified the request of the cellular network and user of the cellular network as service usage activity. *Id.* at 9 (citing Dkt. No. 177-5 at 70:3-71:6).

With respect to Headwater's first contention that service usage activity is a "network request through ConnectivityManager," Defendants respond that Dr. Wesel unequivocally disclaimed this theory during his deposition. Dkt. No. 238 at 4 (citing Dkt. No. 238-4 at 69:14-17 ("The service usage activity that you've identified is the requestNetwork function call itself. Correct? A. No, no, of course not."))). With respect to Headwater's second contention that service usage activity is "requests, packets, or data transmitted to/through the firewall," Defendants contend this is a "red herring," as it is inconsistent with the rest of Headwater's infringement read. *Id.* at 4. Specifically, Defendants claim that in the only instance that Claim 79 and 83 are infringed, there are never requests, packets, or data transmitted to/through the firewall." *Id.* at 4-5.

Headwater responds that Dr. Wesel's opinions are not "siloed" to the "requestNetwork call" or the "firewall." Dkt. No. 258 at 3. Instead, Headwater contends that Android identifies an

app's process as "activity" that requests a network using the requestNetwork call and that transmits data packets through Android's firewall—and that Android blocks the "activity" from doing these things if it is "background activity" having a background process state. *Id.* (citing Wesel Rpt. ¶¶ 380-383). Headwater further contends that Android's "Network Policy manager" classifies whether or not the application is a background activity, and if the application is a background activity, the code will determine instead to disallow the app's network access. *Id.* (citing Wesel Rpt. ¶ 380). Headwater also argues that the Android accused products analyze each request or packet and determine whether to block or allow these communications. *Id.*

Drawing all inferences in favor of Headwater, the Court finds that genuine disputes of material fact exist as to whether the accused Android products identify a service usage activity and determine whether the service usage activity comprises a background activity sufficient to satisfy the limitations in Claim 1. For example, though Defendants claim that the request from the app—*i.e.*, the requestNetwork function call—and the response from the ConnectivityManager are both internal API messages that do not include prospective or successful communications over a wireless network sufficient to be service usage activity, Headwater identifies various ways that the Android products determine whether service usage activity comprises a background activity. These include Headwater's claim that Android makes the determination of whether a service usage activity is a background activity based on information about the activity. *See* Wesel Rpt. ¶¶ 903-04. These also include Headwater's claim that Android's "Network Policy manager" can satisfy Claim 1's limitation because the Network Policy manager classifies an application as a background activity, and if the application is classified as a background activity, then the code will determine instead to disallow the app's network access. *See* Wesel Rpt. ¶¶ 380, 1007. Additionally, there is testimony from Android engineers, such as Jeff Sharkey, that the Android

products block network access requests made by apps running in the background. *See e.g.*, Wesel Rpt. ¶ 904. For at least the foregoing reasons, the Court finds that there are genuine disputes of material fact that make granting summary judgment improper as to the accused Android products.

Accordingly, the Motion for Summary Judgment of Non-Infringement as to the accused Android products on this basis should be **DENIED**.

B. Dependent Claim 79 and 83

Defendants also contend that summary judgment is independently warranted because the accused products do not satisfy the limitations in Claims 79 and 83 requiring "intercepting" or otherwise taking action, such as modifying, blocking, removing, injecting, swapping, or replacing, to disrupt an "application interface message" to "apply the policy" recited in Claim 1. Dkt. No. 177 at 2, 10. Specifically, Defendants assert that though many dependent claims specify different ways to "apply the policy" in Claim 1, including "removing the service usage activity" ('541 Patent at Claim 62) and "informing the first software component whether [it] is allowed to access the wireless network" ('541 Patent at Claim 73), the Asserted Claims take a different approach, namely "intercepting" an API message. *Id.* Defendants claim that no API message is ever "intercepted" or tampered with in the accused products, however, because every API message is received by its intended recipient. *Id.* at 10-11. As the key limitation in these claims involves "intercepting" a message, the Court will begin by addressing the parties' arguments on this issue.

1. The "Intercepting" Limitation in Asserted Claims 79 and 83

Both asserted claims address "intercepting" an API request or message. (*See* '541 Patent, Cl. 79 ("The non-transitory computer-readable storage medium recited in claim 1, wherein apply the policy comprises at least assist in ***intercepting*** a stack application programming interface (API) level or application messaging layer request."); '541 Patent at Claim 83 ("The non-transitory

computer-readable storage medium recited in claim 1, wherein apply the policy comprises at least assist in ***intercepting***, modifying, blocking, removing, injecting, swapping, or replacing an application interface message.") (emphasis added).)

In their Motion, Defendants argue that Headwater's infringement theory for the "intercepting" limitation is precluded by its plain meaning. Dkt. No. 177 at 12. Under the plain meaning, Defendants argue that "a message that is successfully received by its intended recipient is not 'intercepted'" regardless of how the recipient responds because the plain and ordinary meaning of intercepting is "not reaching the intended recipient." *Id.* Defendants analogize the use of "intercepting" to football where a pass is "intercepted" when a member of the opposing team catches the ball, but a pass is not "intercepted" if the pass successfully reaches the team's receiver (*i.e.*, the intended recipient). *Id.*[1]

Defendants further contend that Headwater is precluded from deviating from this ordinary meaning now because Headwater never sought claim construction on "intercepting." *Id.* Even if construction were necessary, Defendants argue that the intrinsic record uses the word "intercept" in the ordinary way. *Id.* (citing '541 Patent at 66:62-67:1 ("[T]he application service interface layer['s] . . . function is to monitor and in some cases intercept and process the traffic between the applications and the standard networking stack API.")),

In its Response, Headwater contends that there is an actual dispute regarding the proper scope of the claim term "intercepting." Dkt. No. 209 at 1-2 (citing *O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1360 (Fed. Cir. 2008) ("When the parties raise an actual

[1] Defendants' use of a football analogy is effective to describe *one* potential ordinary meaning because it conjures up the visual that the intercepted message cannot reach the intended recipient. A counter analogy, however, is when authorities lawfully intercept communications during surveillance activities. Such methods allow the authorities to intercept and listen even though the communication still reaches the intended recipient. Notably, Headwater describes its infringement theory using similar language. *See* Dkt. No. 258 at 5 ("[A] background control policy *listens in* and switches the API response if it is background activity." (emphasis added.)).

dispute regarding the proper scope of these claims, the court, not the jury, must resolve that dispute.")); Dkt. No. 209-2 at ("I guess it comes down to a disagreement about what constitutes an intercepting."). Consequently, in light of the parties' genuine dispute about the meaning of this term, Headwater asks the Court to construe "intercepting." Dkt. No. 209 at 10.

The Court finds that the parties have a genuine dispute as to the scope of the claim term "intercepting." *See also* Pretrial Hearing on June 5, 2025, Rough Tr. at 5:3-5 (Counsel for Defendants requesting an opportunity to present oral argument because the Motion "has a claim construction issue baked in explicitly so by the plaintiff"); *see also id.* at 195:24-25 (Defense counsel describing the "issue" as one of "claim construction"). Accordingly, the Court will resolve that dispute by construing "intercepting." *See O2 Micro*, 521 F.3d at 1362.

Headwater asks the Court to give "intercepting" its plain meaning, which Headwater contends includes (1) "receiving and responding to an API or application messaging layer message or request" and (2) "receiving or observing a message directed to or meant for another." *Id.* at 13. Headwater contends that its proposed construction is consistent with Defendants' arguments to the PTAB in the IPR on this Asserted Patent, testimony from Defendants' experts, and testimony from Dr. Raleigh (the named inventor). *Id.* at 10-13 (citing Defendants' Preliminary Reply in the IPR, Dkt. No. 209-5 at 3 ("Rao's disclosure of communicating the priorities via an API would inform a POSITA that Rao's filter would request the priorities from the agent and the agent would assist in intercepting as claimed, because that is how APIs work")); Wolfe Decl., Dkt. No. 209-7 ¶¶ 256-57 ("A POSITA would further recognize that, in order to perform the callback functions that return the priorities to the filter 322, the agent 326 would need to at least assist in intercepting the filter 322's stack API level request."); Crovella Expert Rpt., Dkt. No. 209-9 ¶ 328 ("The agent must intercept (or assist in intercepting) the filter's stack API level request so that the agent can return

the priorities to the filter."); Crovella Dep. Tr., Dkt. No. 209-10 at 83:11-18 ("Q. And what component in Rao actually intercepts the API-level requests? . . . THE WITNESS: The -- when the filter is sending to the agent, the agent is receiving or intercepting these requests. When the agent is sending to the filter, the filter is receiving or intercepting these requests."); *id.* at 90:21-91:5 ("[W]hen the filter in Rao sends a request to the agent for the agent to transmit the priorities, what's the intended recipient of that request? . . . THE WITNESS: So, in this communication, the recipient is the agent."); Raleigh Dep. Tr., Dkt. No. 209-11 at 154:13-18 (testifying that "intercepting" in claim 79 includes "receiving a stack application program in interface level request or receiving an application messaging layer request"); citing *id.* at 159:5-12 (deposition testimony not included in exhibit).

Headwater also claims that several dictionary definitions do not require that an "intercepted" message be "prevented from reaching its intended destination." Dkt. No. 209 at 12 (citing Merriam-Webster, Dkt. No. 177-10 at 1 ("to receive (a communication or signal directed elsewhere) usually secretly"); Collins Dictionary, Dkt. No. 209-12 ("to see or overhear (a message, transmission, etc., meant for another)."). Accordingly, Headwater states that the construction should "encompass *at least* receiving or observing a message directed to or meant for another." Dkt. No. 209 at 12-13.

Defendants respond that every record dictionary, including Headwater's, confirms that "intercepting" requires "receipt" by someone "other than the intended recipient." Dkt. No. 238 at 5-6. Defendants further contend that it does not matter what happens to the message after the message is intercepted so long as something other than the intended recipient receives it first. *Id.* at 6. Moreover, with respect to the suggestions that Defendants used a different construction in the IPR, Defendants contend that the reference in question explicitly disclosed "intercept[ing] inbound

or outbound network packets [not API messages] associated with applications" before those network packets were transmitted over a network. *Id.* (citing Dkt. No. 238-3 at 2-5). Lastly, Defendants contend that Dr. Raleigh's self-interested testimony "is of little if any significance." *Id.* (citing *Raytheon Co. v. Cray, Inc.*, No. 2:15-CV-1554-JRG, 2017 WL 3034662, at *5 (E.D. Tex. July 18, 2017) ("Inventor testimony, however, is of little if any significance in these claim construction proceedings.").

In its Sur-Reply, Headwater reiterates that Defendants have previously argued that "interception" is met when the intended recipient of an API request receives the request. Dkt. No. 258 at 4. As support, Headwater claims that Defendants contended that "receiving API request" (not the network packets they point to in their Reply) issued directly from a "filter" to an "agent" are "intercepted" by their intended recipient. *Id.* (citing Dkt. No. 209-8 at 23 ("The agent must intercept (or assist in intercepting) the filter's stack API level request . . . ."); Crovella Dep. Tr., Dkt. No. 209-10 at 90:21-91:5 (testifying that the recipient of the filter's request is the agent). Headwater states that this is "not some creative interpretation of Defendants' and their experts' interpretation of 'intercepting'" because the PTAB had the same understanding of Defendants' interpretation. *Id.* (citing Dkt. 258-3 at 18 ("Petitioner argues that agent 326 receiving the filter's requests is 'intercepting a stack application programming interface (API) level . . . request' (claim 79) and 'intercepting . . . an application interface message' (claim 83)." Ultimately, Headwater contends that there is no plausible dispute that the plain meaning of "intercepting" according to the understanding of a POSITA, including Defendants' own experts, encompasses "reception of an API request." Dkt. No. 258 at 4-5.

The Court begins its analysis of the parties' dispute by reviewing the intrinsic record. Here, the specifications use the word "intercept" to describe the function of intercepting traffic between

an app and the standard networking stack API. '541 Patent at 66:62-67:1 ("[T]he application service interface layer['s] . . . function is to monitor and in some cases intercept and process the traffic between the applications and the standard networking stack API."). In other words, the specifications use "intercept" in the ordinary fashion to mean receiving data intended for another. The specifications do not, however, require "intercepting" to mean the message is never delivered to the intended recipient.

Other than the specifications, the parties primarily rely on statements made to the PTAB during the IPR of the asserted patent, dictionary definitions, an expert in a different case, and inventor testimony. Having reviewed the cited evidence, the Court is not persuaded that the statements made to the PTAB during the IPR should carry any weight on the interpretation of "intercepting" in this case. The Court also rejects the contention that testimony by Dr. Crovella, an expert for AT&T in a different case brought by Headwater, carries any real weight on the issue of claim construction in this case. Plaintiff provides no case law support for its contention that these statements should carry significant weight and override the plain and ordinary meaning. The Court also finds that Dr. Raleigh's testimony, as an inventor of the '541 patent, bears reduced weight. *See, e.g., Raytheon Co.*, 2017 WL 3034662, at *5. Accordingly, the Court finds that it would be improper to give "intercepting" the first proposed construction that Headwater offers, which is simply "receiving and responding to an API or application messaging layer message or request."

Having reviewed the parties' positions on the "intercepting" limitation, the Court finds that the appropriate meaning of "intercepting" is "receiving a message directed to or meant for another." This construction does not mean, however, that "intercepting" can only occur if the message "does not reach the intended recipient." The Court notes that this construction is

consistent with various dictionary definitions cited by the parties. *See* Merriam-Webster, Dkt. No. 177-10 at 1 ("to receive (a communication or signal directed elsewhere) usually secretly"); Collins Dictionary, Dkt. No. 209-12 ("to see or overhear (a message, transmission, etc., meant for another)." Having determined that "intercepting" does not require that the message never reach the intended recipient, the Court considers whether there is a genuine dispute of material fact as to infringement of the accused products.

2. The Accused Apple Products

Defendants argue that the accused Apple products do not infringe Claims 79 and 83 because the application interface message is never intercepted or otherwise tampered with and instead, successfully reaches its intended recipient every time. Dkt. No. 177 at 11. Defendants begin by stating that Dr. Wesel confirmed that the API request from the app to the scheduler—*i.e.*, the submitTaskRequest—is the only thing Dr. Wesel accuses as the claimed application interface message for Claims 79 and 83. *Id.* (citing *e.g.*, Dkt. No. 177-5 at 130:18-131:22 ("Q. Are you relying on, for your infringement opinion of Claim 79, any other example of an API level or application messaging layer request other than the submitTaskRequest? A. I don't believe so. I think submitTaskRequest is the one that -- that I am focused on. Q. And same for Claim 83; that submitTaskRequest is the application interface message that you're referring to in the response thereto. Correct? 14 A. Yes, that's correct."). Defendants assert, however, that Dr. Wesel confirmed that the accused request always reaches its intended destination. *Id.* (citing 137:24-138:24 ("Q. Understood. *But there's nothing that prevents the execution of the submitTaskRequest function itself. Correct? A. That's right, nothing prevents the execution of the submitTask function itself*, and that submits a request for a service usage activity, but that service usage activity -- that request, you know, might be intercepted or the service -- or blocked, or whatever, or that service

usage activity might ultimately take place." (emphasis added)). Defendants contend that Headwater makes the "bizarre claim" that sending a message back to the requesting app constitutes "intercepting." *Id.* at 12 (citing Wesel Rpt. ¶¶ 1417-18).

Headwater responds that the "submitTaskRequest" API request that is made by an app is "intercepted" because an app's "submitTaskRequest" API message is "received" by the "BGTaskScheduler API," and the "submitTaskRequest" API's message's "intended destination" is that same "BGTaskScheduler API." Dkt. No. 209 at 13. Headwater then does an about-face and argues that the "intended destination" of the app's "submitTaskRequest" API message is not the "BGTaskScheduler API" at all, but rather the "background task scheduler queue structure." Dkt. No. 209 at 13-14 (citing Wesel Rpt. ¶¶ 1065-1066 (describing Background App Refresh tasks in a queue)). Headwater then argues that the app's request to schedule a Background App Refresh task by invoking the submitTaskRequest method is intercepted because the task that the application requested to be placed into the task scheduler queue is actually not placed into the queue. *Id.* at 14 (citing Wesel Rpt. ¶¶ 1418-1422 ("The interception occurs when _DASDaemonClient.m prevents the activity from getting submitted to _DASDaemon.m . . . .")). Put another way, Headwater argues that iOS's "prevention" or "denial" of the barActivity object or Background App Refresh task request, which is intended by an app to be placed into the scheduler queue, constitutes infringement because the requested task never reaches its intended destination in the scheduler queue. Dkt. No. 209 at 14; Dkt. No. 258 at 5, n.6.

Defendants respond that Headwater's assertion that the "intended destination" is not the "BGTaskScheduler API" is false because Dr. Wesel explained that the "submitTaskRequest" call is intended to reach and does reach the "BGTaskScheduler API," which always responds. Dkt. No. 238 at 6 (citing ¶¶ 274, 1418, Dkt. No. 177-5 at 131:23-132:21; 137:24-138:24). Defendants also

argue that "denying" a request such that it is not placed into the queue is not "intercepting" nor is it modifying, blocking, removing, injecting, swapping, or replacing an application interface message because nothing changes the way messages are sent in the normal-course API exchanges that Headwater accuses. Dkt. No. 238 at 6-7; Dkt. No. 177 at 11 (citing Dkt. No. 177-3 at 213:14-17 ("[D]oes background app refresh intercept, modify, block, remove, inject, swap, or replace API requests from applications? A. No, absolutely not.")).

Headwater responds that Dr. Wesel never opined that the intended destination of a Background App Refresh task request is the "BGTaskScheduler API." Dkt. No. 258 at 5, n.6. Having reviewed the citations provided by Defendants, the Court agrees with Headwater. Dr. Wesel never opines that the intended destination is the "BGTaskScheduler API." Instead, Dr. Wesel opines that the requests are sent to "DASScheduler.m," which Headwater claims is iOS's DASDaemon service that maintains the queue structure and adds the Background App Refresh tasks into the queue. Dkt. No. 209 at 14; Wesel Rpt. ¶ 1419. Headwater further claims that interception occurs when "_DASDaemonClient.m" prevents the activity from getting submitted to "_DASDaemon.m" if "barEnabled" is false and the app is in the disallowed app list. Dkt. No. 209 at 14 (citing Wesel Rpt. ¶¶ 1418-19). Putting it another way, Headwater contends that the accused products have an API that receives API requests and that blocks the requests if it is for background activity, because a "background control policy listens in and switches the API response if it is background activity." Dkt. No. 258 at 5.

Based on the foregoing arguments, the Court finds that there are several factual disputes, including for example a dispute as to what constitutes the intended destination of the Background App Refresh task requests and whether "_DASDaemonClient.m" preventing the activity from getting submitted to "_DASDaemon.m" or to the "background task scheduler queue structure"

meets the "intercepting" limitation in Claims 79 and 83. For at least these reasons, summary judgment of non-infringement is not proper as to the accused Apple products.

Accordingly, the Motion on this basis should be **DENIED**.

3. The Android Accused Products

Defendants similarly contend that the accused Android products do not infringe Claims 79 and 83 because the accused application interface messages Dr. Wesel accuses always reach their intended recipients. Dkt. No. 177 at 13. Specifically, Defendants claim that Dr. Wesel accuses the "requestNetwork" call from the app and the "response back to the app from the ConnectivityManager" as the only messages accused as the claimed application interface message for Claims 79 and 83. *Id.* at 13 (citing Dkt. No. 177-5 at 93:9-94:15; 107:13-108:13). Defendants further claim that the "requestNetwork" call from the app, which is received by the ConnectivityManager, and the "response back to the app from the ConnectivityManager," which is received by the app, are always successfully delivered to their respective intended recipients. *Id.* at 13-14 (citing *e.g.*, Dkt. No. 177-5 at 105:8-18 ("Q. Okay. Is there any entity other than the ConnectivityManager that could receive the requestNetwork request? A. No, the request is processed by the ConnectivityManager. Q. Regardless of whether the response is ultimately to provide a network or to not provide a network, the ConnectivityManager receives that request. Correct? A. It is the ConnectivityManager that either intercepts the request and denies the network or that will allow the application to have the network.")).

Headwater responds that the accused Android products infringe Claims 79 and 83 because the network policy rules result in the app's request for network access being blocked or denied and then told that the network is unavailable. Dkt. No. 209 at 15. Headwater claims support in Defendants' expert's confirmation that the "intercepting" does not require "blocking." Dkt. No.

209-2 at 101:4-6. Headwater further argues that Dr. Wesel explains that the "Network Policy rules assist in intercepting an application's network request to the Connectivity Manager application programming interface, which will indicate to the requesting application that an available network is DISCONNECTED / BLOCKED if it is background activity." Dkt. No. 209 at 15. Headwater claims this is sufficient on its own to warrant denial of Defendants' summary judgment motion. Dkt. No. 209 at 15.

Even if it is not sufficient to warrant denial of the Motion, Headwater argues that other genuine disputes of material fact exist. *Id*. Specifically, Headwater argues that when Android applies the accused policies, those policies result in the normal procedure for application interface messages being disrupted or network access being denied to the app and told that the network is unavailable. *Id*. (citing Wesel Rpt. ¶ 1293 ("Instead of reaching an available network with its API level request for the network, the application is denied network access and told that the network is unavailable.")).

Defendants, however, dispute that the "normal procedure" for API messages is "disrupted" when the "application is denied network access and told that the network is unavailable" because, as Defendants contend, this "is precisely the normal operation of the ConnectivityManager API, which must successfully receive an application's network request to be able to respond that 'the network is unavailable' at all." Dkt. No. 238 at 7. Defendants further reiterate their contention that the ConnectivityManager API always receives the requestNetwork call and always provides a response to the application while no other component of Android receives or otherwise interferes with either message. *Id.*

In its Sur-Reply, Headwater argues only that the request for the network is intercepted and blocked when the ConnectivityManager API receives the API request and, if it is for background

activity, responds that the network is unavailable due to Android's background control policy. Dkt. No. 258 at 5.

The Court similarly finds that summary judgment of non-infringement is not proper as to the accused Android products. Though Defendants contend that the "requestNetwork" call from the app, which is received by the ConnectivityManager, and the "response back to the app from the ConnectivityManager," which is received by the app, are always successfully delivered to their respective intended recipients, the Court has construed "intercepting" such that there is no requirement that the message never reach the intended recipient. This alone prevents summary judgment of non-infringement. Nevertheless, the Court also finds that Headwater identifies the "Network Policy rules" as the way in which the "intercepting" occurs. *See* Wesel Rpt. ¶ 1291 ("For example, Network Policy rules assist in intercepting an application's network request to the Connectivity Manager application programming interface, which will indicate to the requesting application that an available network is DISCONNECTED / BLOCKED if it is background activity."). Dr. Wesel also opines that when Android applies the accused policies, those policies result in the normal procedure for application interface messages being disrupted or network access being denied to the app and told that the network is unavailable. *Id*. (citing Wesel Rpt. ¶ 1293 ("Instead of reaching an available network with its API level request for the network, the application is denied network access and told that the network is unavailable.")). This sufficiently raises a fact question as to whether Claim 83's additional limitations—*i.e.*, modifying, blocking, removing, injecting, swapping, or replacing an application interface message—are met. For at least these reasons, summary judgment of non-infringement is not proper as to the accused Android products.

Accordingly, the Motion on this basis should be **DENIED**.

IV. CONCLUSION

For the reasons discussed above, it is recommended that the Motion for Summary Judgment of Non-Infringement of U.S. Patent No. 8,589,541 be **DENIED**.

A party's failure to file written objections to the findings, conclusions and recommendations contained in this report **by not later than June 18, 2025** bars that party from *de novo* review by the District Judge of those findings, conclusions, and recommendations and, except on grounds of plain error, from appellate review of unobjected-to factual findings and legal conclusions accepted and adopted by the district court. FED. R. CIV. P. 72(b)(2); *see also Douglass v. United Servs. Auto. Ass'n*, 79 F.3d 1415, 1428–29 (5th Cir. 1996) (*en banc*). Any objection to this Report and Recommendation must be filed in ECF under the event "Objection to Report and Recommendation [cv, respoth]" or it may not be considered by the District Judge.

SIGNED this 6th day of June, 2025.

ROY S. PAYNE
UNITED STATES MAGISTRATE JUDGE